REMARKS

Double patenting

Claims 1-20 have been provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being obvious and unpatentable over claims 1-20 of copending US patent application no. 10/679,076. Applicant first notes that insofar as this rejection is provisional – i.e., the patent application 10/679,076 has not yet issued as a patent – and that insofar as there are pending prior art rejections in the present patent application, filing a terminal disclaimer at this junction is premature, although Applicant reserves the right to file a terminal disclaimer later in the prosecution of this application.

However, second, Applicant respectfully traverses this obviousness-type double patenting rejection. The Examiner has stated that although "the conflicting claims are not identical, they are not patentably distinct from each other because both use either a software or a hardware approach to monitor (locking) memory." (Office action of July 5, 2007, p. 3, para. 6.) Applicant respectfully disagrees with the Examiner's analysis in this respect.

The MPEP's precepts as to obviousness rejections in general is relevant to the present inquiry as to whether there is obviousness-type double patenting as to the present patent application vis-à-vis patent application 10/679,076. The MPEP states that "*all* the claim limitations must be taught or suggested by the prior art." (MPEP sec. 2143.03, citing In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)) "*All* words in a claim must be considered in judging the patentability of that claim against the prior art." (Id., citing In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)) That is, "the claimed invention as a whole must be considered." (MPEP sec. 2141.02.I.) "Distilling an invention down to the 'gist' or 'thrust' of an invention disregards the requirement of analyzing the subject matter 'as a whole.'" (MPEP sec 2141.02.II., citing W.L. Gore & Associates, Inc. v. Garlock, Inc., 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983))

In stating that the claims of the present patent application are patentably indistinct from the claims of patent application 10/679,076, Applicant therefore respectfully submits that the Examiner has distilled both the invention of the present patent application and the invention of patent application 10/679,076 to their "gists," without considering the claimed inventions as a whole, and without considering all the words in the claims of these applications. In other words, these patent applications are both directed to claimed inventions that are much more than simply using a software or a hardware approach to locking memory.

For example, consider claim 1 of the present patent application. Claim 1 of the present patent application is limited to a software approach to locking memory being utilized to execute a code section, and a pseudo-transaction is employed to determine whether a hardware approach to transactional memory to execute the code section would have been successful. Where this hardware approach satisfies a threshold based on success of at least the pseudo-transaction, the hardware approach is subsequently used to execute the code section.

By comparison, consider claim 1 of patent application 10/679,076, as presently pending. Claim 1 of patent application 10/679,076 is limited to executing a code section using a hardware approach to transactional memory. Where the hardware approach fails a threshold in executing the code section, the code section is instead executed using a software approach to locking memory.

Thus, in the present patent application, you first utilize a software approach to locking memory to execute a code section, and only if a hardware approach to transactional memory satisfies a threshold based on success of at least a pseudo-transaction, do you then subsequently use the hardware approach to execute the code section. By comparison, in patent application 10/679,076, you first use a hardware approach to transactional memory to execute the code section, and only if this hardware approach fails a threshold do you use a software approach to locking memory to execute the code section. That is, in the present patent application, you first utilize a software approach, and then subsequently utilize a hardware approach if it is determined that the hardware approach satisfies a threshold based on success of a pseudo-transaction. By comparison, in patent application 10/679,076, you first use a hardware approach – not a software approach – and *then* use a software approach if the hardware approach has failed. This difference in the order of usage of the software and the hardware approaches is one patentable distinction between these two patent applications.

A second patentable distinction between these two patent applications is that in the present patent application, if the hardware approach *satisfies* a threshold then you subsequently use the hardware approach, whereas in patent application 10/679,076, if the hardware approach *fails* a threshold then you instead use the software approach. A third patentable distinction between these two patent applications is that the present patent application employs a *pseudo-transaction* to determine whether the hardware approach *would have* been successful, whereas patent application 10/679,076 is silent as to employing a *pseudo-transaction*, and is in fact silent as to determining whether the hardware approach *would have* been successful. For all of these reasons, therefore, Applicant submits that the present patent application is patentably distinct from patent application 10/679,076.

Prior art rejections as to claims 1-10

Claim 1 is an independent claim, from which claims 2-10 ultimately depend. Claims 1-10 have been rejected under 35 USC 103(a) as being unpatentable over Slingwine (6,816,952) in view of Vartti (6,219,690). Applicant respectfully submits that as previously presented, claim 1 is patentable over Slingwine in view of Vartti, such that claims 2-10 are patentable at least because they depend from a patentable base independent claim, claim 1.

It is noted that "[t]o establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art." (In re Royka, 490 F.2d 981, 180 USPQ 580 (CCPA 1974)) "All words of a claim must be considered in judging the patentability of that claim against the prior art." (In re Wilson, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA

1970)) Applicant respectfully submits that not all the claim limitations of claim 1 are taught or suggested by Slingwine in view of Vartti. Applicant now presents a number of the limitations of claim 1, and discusses how these limitations are not taught or suggested by Slingwine in view of Vartti.

Execution of a code section

Throughout claim 1, a code section is said as being executed. Thus, in the first element, a software approach to locking memory is utilized "to execute a code section." Likewise, in the second element, a hardware approach to transactional memory is utilized "to execute the code section."

Applicant submits, however, that Slingwine in view of Vartti does not teach, disclose, or suggest execution of a *code section*. As to Slingwine, the Examiner appears to have stated that this reference discloses in column 4, lines 6-67, and column 5, lines 1-28, the second element of claim 1,[1] which specifically recites "where the hardware approach to transactional memory to

[1] Applicant says here that "the Examiner appears to have stated" that Slingwine discloses the second element of claim 1 in column 4, lines 6-67, and column 5, lines 1-28, because paragraph 9 on page 5 of the office action of July 5, 2007, actually states in relevant part that:

> Regarding claim 1, Slingwine et al., discloses a method comprising: [second element of claim 1], however Vartti et al. does (col. 4, lines 6-67 and col. 5 lines 1-28).

Applicant believes that the Examiner mistakenly added "Vartti et al." to this paragraph. This is because the Examiner found the first element of claim 1 in Vartti, as discussed on page 6 of the office action. Therefore, if both the first and the second elements of claim 1 were found in Vartti, then the Examiner would have to have rejected claim 1 under 35 USC 102 as to Vartti alone, not under 35 USC 103 as to Slingwine in view of Vartti. <u>However, Applicant respectfully requests clarification as to whether the Examiner is relying upon column 4, lines 6-67 and column 5, lines</u>

execute the code section satisfies a threshold based on success of at least the pseudo-transaction, subsequently utilizing the hardware approach to transactional memory to execute the code section." However, nowhere in these portions of Slingwine, nor other portions of Slingwine, does this reference actually ever disclose execution of a code section by utilizing a hardware approach to transactional memory (or a software approach to locking memory). Rather, Slingwine is focused on *accessing data*, not *executing a code section*. For instance, Slingwine states that "memory data are accessed sequentially." (Col. 4, ll. 9-10.) Slingwine states that "[w]hen *accessed* by one of processors 16, the shared *data* are protected by software locks." (Col. 4, ll. 33-35.) Slingwine states that "[t]o alleviate the obvious problem of increased *data access* time, the shared data structures may be stored in cache memory." (Col. 4, ll. 36-38.) Thus, Slingwine does not employ either a hardware or a software approach *to execute a code section*, as in the claimed invention, but rather employs its described approach *to access data*.

As to Vartti, the Examiner has stated that this reference discloses in column 2, lines 58-67 and in column 3, lines 1-28 and 44-62, the first element of claim 1, which specifically recites "utilizing a software approach to locking memory to execute a code section relating to memory and employing a pseudo-transaction to determine whether a hardware approach to transactional memory to execute the code section would have been successful." (Office action, p. 6.) However, nowhere in these portions of Vartti, nor other portions of Vartti, does this reference actually ever disclose execution of a code section by utilizing a software approach to locking memory (or a hardware approach to transactional memory). As with Slingwine, Vartti is focused on *accessing or retrieving data*, not *executing a code section*. For instance, Vartti states that a "thread then 'unlocks', or deactivates sole *access* rights to the *data*." (Col. 2, ll. 13-14.) Vartti

1-28, of *Slingwine* or of *Vartti*, in the next office action. In any case, whether the Examiner has relied upon Slingwine or Vartti as teaching various limitations of claim 1 is in fact moot, because neither Slingwine or Vartti disclose these limitations, as is discussed in the body of this office action response.

states that "the shared *data* must not be *accessed* without first gaining authorization." (Col. 2, ll. 34-35.) Vartti states that a "drawback associated within prior art software-locking mechanisms involves the time associated with *retrieving* software-lock-protected *data*." (Col. 3, ll. 13-15.) Vartti states that "one or more addresses in the lock cell cache line are used as pointers to *retrieve* cache lines," where a cache line is "a cache line of *data*." (Col. 3, ll. 44-46; col. 3, l. 62.) Thus, Vartti also does not employ either a hardware or a software approach *to execute a code section*, as in the claimed invention, but rather employs its described approach *to access data*.

Therefore, because Slingwine in view of Vartti does not teach, disclose, or suggest execution of a code section by utilizing a software approach or a hardware approach, not all the claim limitations are taught, disclosed, or suggested by Slingwine in view of Vartti. As such, the claimed invention is patentable over Slingwine in view of Vartti for at least this reason.

Pseudo-transaction

In claim 1, a *pseudo-transaction* is employed to determine whether a hardware approach to transactional memory to execute the code section would have been successful. The online dictionary www.dictionary.com relevantly defines the adjective *pseudo* as meaning "not actually but having the appearance of." Thus, one of ordinary skill within the art would accord the term pseudo-transaction with its ordinary and customary meaning, as not actually being a transaction, but having the appearance of a transaction, consistent with the usage of this term in the patent application as filed, such as in paragraph [0029].

Applicant submits, however, that Slingwine in view of Vartti does not teach, disclose, or suggest usage of a pseudo-transaction in any way whatsoever, let alone to specifically determine whether a hardware approach to execute the code section would have been successful. For instance, in column 4, lines 6-67 and column 5, lines 1-28 of Slingwine, as well as in Slingwine as a whole, this reference does not refer to a pseudo-transaction, and does not use any type of transaction consistent with the ordinary and customary meaning of this term as not actually being a transaction, but having the appearance of a transaction. Slingwine simply discusses using preventive or corrective solutions to data coherence problems, where "[p]reventive solutions typically use software to maintain data coherence while corrective solutions typically use hardware for detecting and resolving data coherence problems." (Col. 4, ll. 25-28.) Neither of these solutions, however, is described as employing a *pseudo-transaction* in any way.

Likewise, in column 2, lines 58-67 and column 3, lines 1-28 and 44-62 of Vartti, as well as in Vartti as a whole, this reference does not refer to a pseudo-transaction, and does not use any type of transaction consistent with the ordinary and customary meaning of this term as not actually being a transaction, but having the appearance of a transaction. Vartti simply discusses using "[v]arious types of locking mechanisms," which may "use a lock cell or a semaphore." (Col. 2, ll. 14-16.) However, none of the locking mechanisms disclosed in Vartti are described as employing a *pseudo-transaction* in any way. Therefore, because Slingwine in view of Vartti does not teach, disclose, or suggest employing a *pseudo-transaction* in any way, not all the claim limitations are taught, disclosed, or suggested by Slingwine in view of Vartti. As such, the claimed invention is patentable over Slingwine in view of Vartti for at least this reason as well.

Determining whether a hardware approach <u>would have been successful</u>

In claim 1, it is determined whether a hardware approach to transactional memory *would have been successful* to execute a code section, specifically by employing a pseudo-transaction, where a software approach to locking memory has actually been utilized to execute the code section in question. It is important to understand the usage of the verb phrase "would have been successful" as employed in claim 1. In the first element of claim 1, a software approach is in actuality utilized to execute a code section. However, this first element is further limited to determining whether a hardware approach *would have been* successful in executing the code section. That is, the hardware approach was *not actually used* to execute the code section in the first element of claim 1 — rather, the software approach was actually used to execute the code section. Instead, what this first element of claim 1 determines is whether the hardware approach, if it had been used, *would have been* successful; not, for instance, whether the hardware approach *was* successful.

This usage of the phrase "would have" is described well at the Internet web page grammar.ccc.commnet.edu/grammar/conditional.htm, which discusses in relevant part:

> For **past unreal** events — things that didn't happen, but we can imagine — we put the verb in the condition clause a further step back — into the past perfect:
>
> If the Pacers had won, Aunt Glad would have been rich.
>
> If she had bet that much money on the Bulls, she and Uncle Chester could have retired.
>
> I wish I had lived in Los Angeles when the Lakers had Magic Johnson.
>
> If I had known you were coming, I would have baked a cake.
>
> In this last sentence, note the conditional clause in the past perfect (had known) and the result clause that uses the conditional modal + have + the past participle of the main verb (would have baked).

Thus, determining whether the hardware approach *would have* been successful in executing the code section as employed in the first element of claim 1 means the following. Specifically, this element of claim 1 determines – if the software approach had not been utilized to execute the code section (when in fact it has been explicitly utilized), and if the hardware approach had instead been utilized to execute the code section (when in fact it has not been utilized) – whether the hardware approach would have been successful. As such, consistent with the Internet web page excerpted above, the utilization of the hardware approach to execute the code section is a "past unreal event" – the hardware approach was not actually used in the first element of claim 1, but rather it is determined that if the hardware approach had been used, whether the hardware approach *would have been successful*.

Therefore, Applicant submits that Slingwine in view of Vartti does not determine whether a hardware approach to execute a code section *would have been successful*. The Examiner has stated that Vartti discloses determining whether a hardware approach to transactional memory to execute the threshold *would have been successful*, specifically in column 2, lines 58-67, and column 3, lines 1-28 and 44-62 thereof. However, Vartti does not even disclose employing a hardware approach to transactional memory in this and other portions of the reference, let alone disclosing determining whether such a hardware approach *would have been successful*. Rather, Vartti is focused on employing a software approach to locking memory. For instance, Vartti discloses that "[v]arious types of *locking* mechanism have been introduced." (Col. 2, ll. 15-16.) In particular, Vartti discloses that you "control a *software-lock*." (Col. 2, l. 18; see also, col. 2, ll. 22, 24-25, 32, 35, 51, 61.) The discussion of such a locking mechanism employing a software-lock in Vartti cannot be considered as determining whether a hardware approach to transactional memory *would have been successful* in executing the code section if the hardware approach had been employed instead of the software approach.

Therefore, because the reference – Vartti – that the Examiner has relied upon in teaching this limitation of the claimed invention does not actually disclose the limitation, Slingwine in view of Vartti does not teach, disclose or suggest all the limitations of claim 1. As such, the claimed invention is nonobvious and patentable over Slingwine in view of Vartti for at least this reason, too.

Prior art rejections as to claims 11-16

Claim 11 is an independent claim, from which claims 12-16 ultimately depend. The Examiner has stated that claims 11-16 have been rejected under 35 USC 103(a) as being unpatentable over Slingwine in view of Vartti. (Office action of July 5, 2007, p. 5, para. 9.) However, in specifically discussing the rejection of claims 11-14 and 16, where claims 12-14 and 16 ultimately depend from claim 11, the Examiner has explained how he believes that *all* the limitations of these claims are specifically *disclosed* by Slingwine alone. (Id., pp. 9-11.) Therefore, as an initial matter, Applicant submits that the Examiner has incorrectly rejected claims 11-14 and 16 under 35 USC 103 as being unpatentable over Slingwine in view of Vartti. Where *all* the limitations of the claims have been stated as being specifically *disclosed* in a single reference – Slingwine – the proper basis of rejection is anticipation by Slingwine under 35 USC 102. Applicant therefore requests clarification in this respect in the next office action.

Therefore, in the present office action response, Applicant treats claims 11-14 and 16 as being rejected under 35 USC 102 as being anticipated by Slingwine, and claim 15 as being rejected under 35 USC 103 as being unpatentable over Slingwine in view of Vartti, consistent with the Examiner's *reasoning* in rejecting these claims. Applicant respectfully submits that as originally presented, claim 11 is patentable over Slingwine, such that claims 12-16 are patentable at least because they depend from a patentable base independent claim, claim 11.

Applicant respectfully reminds the Examiner that it is the Examiner's obligation to present *prima facie* anticipation, not Applicant's. "It is by now well settled that the burden of establishing a *prima facie* case of anticipation resides with the Patent and Trademark Office." (In re Skinner, 2 USPQ2d 1788 (BPAI 1986)) The Federal Circuit expects the Patent and Trademark Office's "anticipation analysis [to] be conducted on a limitation by limitation basis, with specific fact findings for each contested limitation and *satisfactory explanations* for such findings." (Gechter v. Davidson, 43 USPQ2d 1030 (Fed. Cir. 1997) (emphasis added))

In the present office action, however, the Examiner has failed his obligation to present *prima facie* anticipation. The Examiner has simply recited all the limitations of claim 11, and then has summarily stated that specific portions of Slingwine disclose all these limitations, without any explanation whatsoever. For instance, the sum total of the Examiner's rejection of independent claim 11 is as follows: "Regarding claim 11, Slingwine et al., discloses [claim 11 recited in its entirety] (col. 2 lines 58-67 and col. 3 lines 1-28 and 44-62." As such, the Examiner has ignored the Federal Circuit's requirement that "anticipation analysis be conducted on a *limitation by* *limitation basis*" (i.e., not on a "claim by claim" basis). Likewise, while the Examiner has seemingly provided fact findings for claim 11 as a whole (i.e., by reciting specific lines of columns 2 and 3 in Slingwine), he has failed to provide *specific* fact findings *for each limitation* – and has not provided *any* explanation of these findings, let alone *satisfactory* explanations, as is required.

In any case, it is noted that, under 35 USC 102, every limitation of a claim must *identically* appear in a single prior art reference for it to anticipate the claim. (In re Bond, 15 USPQ2d 1566 (Fed. Cir. 1990)) That is, the standard for anticipation under 35 USC 102 is that "[t]here must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention." (Scripps Clinic & Research Found. v. Genentech, Inc., 18 USPQ2d 1001, 1010 (Fed. Cir. 1991)) The prior art reference must disclose each element of the claimed invention "arranged as in the claim" in question. (Lindermann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 221 USPQ 481, 485 (Fed. Cir. 1984)) Applicant respectfully submits that not all the claim limitations are disclosed in Slingwine. Applicant specifically presents a number of the claim limitations of claim 11, and discusses how these limitations are not disclosed in Slingwine.

Processor having transactional memory capability

Claim 11 is limited to a "processor having transactional memory capability." As with all the limitations of claim 11, the Examiner has stated that this limitation is found in Slingwine in column 2, lines 58-67, and column 3, lines 1-28 and 44-62. However, column 2, lines 58-67, just note that a particular reference "describes single processor and multiprocessor computer implementations of a UNIX® operating system in which a process 30 has an asleep state 32, a ready to run state 34, a kernel running state 36, and a user running state 38." The processor of such a system is not described in Slingwine as having transactional memory capability. Column 3, lines 1-28, just note another particular reference that "describe[s] a presence flag mutual-exclusion technique," relating to "cache memory controllers . . . and a shared memory controller."

This portion of Slingwine does not even disclose a processor, let alone a processor having transactional memory capability. Column 3, lines 44-62 discusses single path, single data (SPSD) systems that “include multiprocessor systems that time-share a single bus or use a crossbar switch,” where in “such systems, the processors do not have cache memories.” The processor of such a system is not described in Slingwine as having transactional memory capability. Applicant has in fact reviewed Slingwine in its entirety, and can find no reference to a processor having transactional memory capability.

Therefore, because Slingwine does not disclose a processor having transactional memory capability, Slingwine cannot anticipate claim 11.

Pseudo-transactional memory capability

Claim 11 is limited to the processor having memory capability such that it includes “a pseudo-transactional memory capability.” The adjective *pseudo* has been discussed above in relation to claim 1, and the meaning of pseudo-transaction, and thus the meaning of pseudo-transactional, is not repeated here to avoid redundancy. As with all the limitations of claim 11, the Examiner has stated that this limitation is found in Slingwine in column 2, lines 58-67, and column 3, lines 1-28 and 44-62. However, in these portions of Slingwine, this reference does not refer to pseudo-transactional memory capability, and does not use any type of transaction consistent with the ordinary and customary meaning of this term. Slingwine simply discusses “single processor and multiprocessor computer” system in column 2, lines 58-67; “a presence flag mutual-exclusion technique” relating to “cache memory controllers and a shared memory controller” in column 3, lines 1-28; and, single path, single data (SPSD) systems that “include multiprocessor systems that time-share a single bus or use a crossbar switch” where “the processors do not have cache memories” in column 3, lines 44-62. Applicant has reviewed Slingwine in its entirety, and, as it is understood by Applicant, can find no reference to pseudo-transactional memory capability.

Therefore, because Slingwine does not disclose pseudo-transactional memory capability, Slingwine cannot anticipate claim 11 for this reason as well.

Determine whether the transactional memory capability would have been successful

In claim 11, the pseudo-transactional memory capability "determines whether the transactional memory capability *would have been successful*." Interpretation of the claim limitations "would have been successful" has been discussed above in relation to claim 1, and is not repeated here to avoid redundancy. As with all the limitations of claim 11, the Examiner has stated that this limitation is found in Slingwine in column 2, lines 58-67, and column 3, lines 1-28 and 44-62. However, in these portions of Slingwine, this reference does not determine whether a transactional memory capability *would have been successful*. Again, Slingwine simply discusses "single processor and multiprocessor computer" system in column 2, lines 58-67; "a presence flag mutual-exclusion technique" relating to "cache memory controllers and a shared memory controller" in column 3, lines 1-28; and, single path, single data (SPSD) systems that "include multiprocessor systems that time-share a single bus or use a crossbar switch" where "the processors do not have cache memories" in column 3, lines 44-62. Applicant has reviewed Slingwine in its entirety, and, as it is understood by Applicant, can find no reference to determining whether any type of capability – let alone a transactional memory capability of a processor – *would have been successful*.

Therefore, because Slingwine does not disclose determining whether a transactional memory capability would have been successful, Slingwine cannot anticipate claim 11 for this reason, too.

Execution of a code section

In claim 11, a spin lock function is to "execute a code section." As with all the limitations of claim 11, the Examiner has stated that this limitation is found in Slingwine in column 2, lines

58-67, and column 3, lines 1-28 and 44-62. However, in these portions of Slingwine, this reference does not actually ever employ a spin lock function to specifically execute a code section. Again, Slingwine simply discusses "single processor and multiprocessor computer" system in column 2, lines 58-67; "a presence flag mutual-exclusion technique" relating to "cache memory controllers and a shared memory controller" in column 3, lines 1-28; and, single path, single data (SPSD) systems that "include multiprocessor systems that time-share a single bus or use a crossbar switch" where "the processors do not have cache memories" in column 3, lines 44-62. Applicant has reviewed Slingwine in its entirety, and can find no reference to any type of function – let alone a spin lock function – that *executes a code section*.

Therefore, because Slingwine does not disclose a spin lock function to execute a code section, Slingwine cannot anticipate claim 11 for this reason as well.

Prior art rejections as to claims 17-20

Claim 17 is an independent claim, from which claims 18-20 ultimately depend. Claims 17-20 have been rejected under 35 USC 103(a) as being unpatentable over Slingwine in view of Vartti. Applicant respectfully submits that as originally presented, at least claim 17 is patentable over Slingwine in view of Vartti, such that claims 18-20 are patentable at least because they depend from a patentable base independent claim, claim 17.

Applicant respectfully submits that not all the claim limitations of claim 17 are taught or suggested by Slingwine in view of Vartti. Applicant now presents a number of the claim limitations of claim 17, and discusses how these limitations are not taught or suggested by Slingwine in view of Vartti. Applicant's discussion in this respect substantially summarizes the arguments presented above in relation to claim 1, insofar as the Examiner has stated that claim 17 has been rejected "under the same rationale as claim 1." (Office action of July 5, 2007, p. 12.)

Execution of a code section

Throughout claim 17, a code section is said as being executed. Thus, the means utilizes a hardware approach to transactional memory "to execute a code section" after having utilized a software approach to locking memory "to execute the code section." Applicant submits, however, that Slingwine in view of Vartti does not teach, disclose, or suggest execution of a code section. Column 4, lines 6-67 and column 5, lines 1-28 of Slingwine, as relied upon by the Examiner in rejecting claim 1 and hence claim 17 as well, do not disclose execution of a code section, as has been discussed in detail above in relation to claim 1. Likewise, column 2, lines 58-67 and column 3, lines 1-28 and 44-62 of Vartti, as relied upon by the Examiner in rejecting claim 1 and hence claim 177 as well, do not disclose execution of a code section, as has been discussed in detail above in relation to claim 1. Therefore, where neither Slingwine nor Vartti teaches execution of a code section, Slingwine in view of Vartti does not teach, disclose, or suggest all the claim limitations of claim 17, and claim 17 is patentable and nonobvious over Slingwine in view of Vartti.

Pseudo-transaction

In claim 17, a *pseudo-transaction* is employed to determine whether a hardware approach to transactional memory would have been successful in executing the code section. The meaning of the adjective *pseudo* has been discussed in detail in relation to claim 1 above. Applicant again submits that Slingwine in view of Vartti does not teach, disclose, or suggest any type of usage of a pseudo-transaction. Neither Slingwine nor Vartti refer to a pseudo-transaction in any portion thereof, nor does either reference use any type of transaction consistent with the ordinary and customary meaning of this term, as has also been discussed in detail above in relation to claim 1. Therefore, where neither Slingwine nor Vartti teaches utilization of a pseudo-transaction as particularly employed in claim 17, Slingwine in view of Vartti does not teach, disclose or suggest all the claim limitations of claim 17, and claim 17 is patentable and nonobvious over Slingwine in view of Vartti.

Determining whether a hardware approach would have been successful

In claim 17, it is determined whether a hardware approach to transactional memory *would have been successful* in executing a code section, specifically by employing a pseudo-transaction, after a software approach to locking memory has already been utilized to execute the code section in question. The meaning of the verb phrase "would have been" has been discussed in detail in relation to claim 1 above. Applicant again submits that Slingwine in view of Vartti does not teach, disclose, or suggest any determination as to whether a hardware approach *would have been successful* in executing a code section after a software approach has already been utilized to execute the code section, as has also been discussed in detail above in relation to claim 1. Therefore, where neither Slingwine nor Vartti teaches determining whether a hardware approach *would have been successful* in executing a code section, Slingwine in view of Vartti does not teach, disclose, or suggest all the claim limitations of claim 17, and claim 17 is patentable and nonobvious over Slingwine in view of Vartti.

Conclusion

Applicants have made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Mike Dryja, Applicants' Attorney, at 425-427-5094, so that such issues may be resolved as expeditiously as possible. For these reasons, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,

October 5, 2007
Date

Michael A. Dryja, Reg. No. 39,662
Attorney/Agent for Applicant(s)

Law Offices of Michael Dryja
1474 N Cooper Rd #105-248
Gilbert, AZ 85233
tel: 425-427-5094
fax: 425-563-2098